2022-12-04 Final RF Models

Julie Ye

Sunday, December 4, 2022

Model Fitting

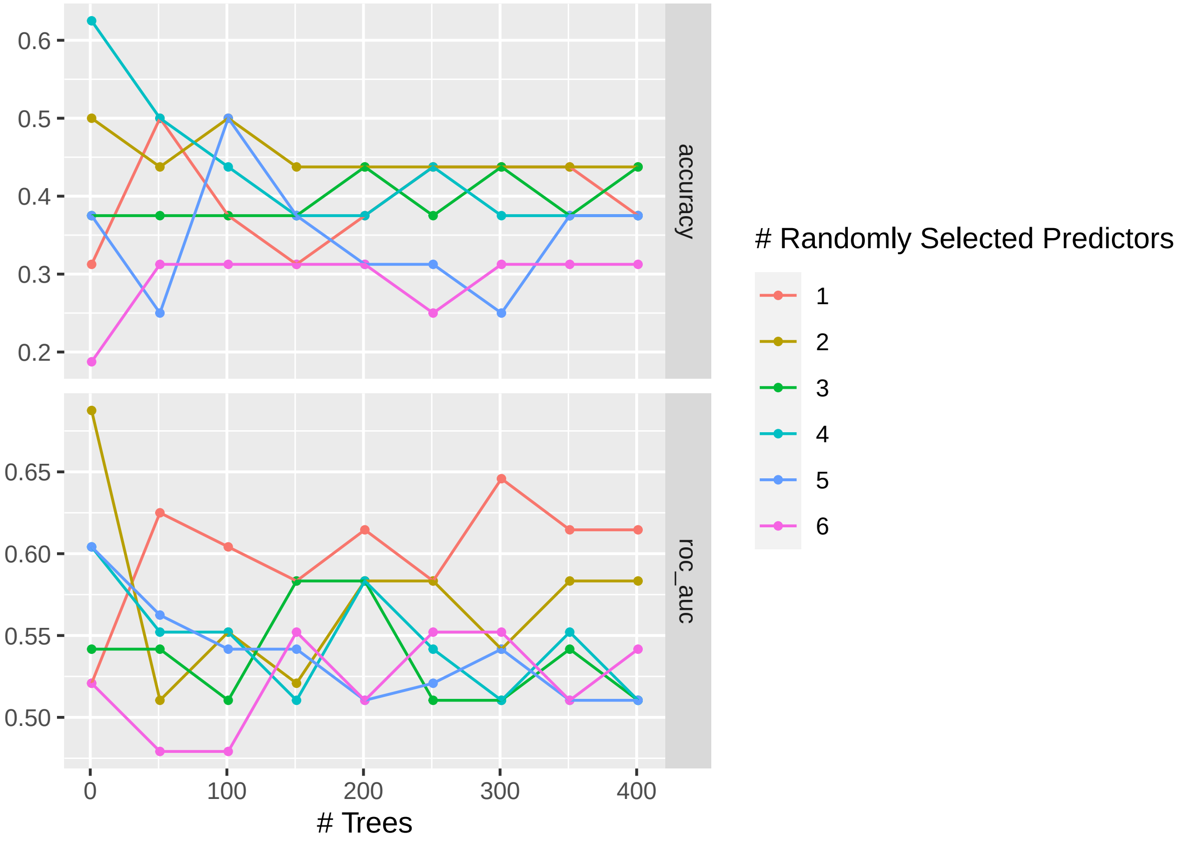
There are two rows of observations (statistics) per match, corresponding to the home team and away team. For our models, we have divided the UEFA Women's EURO 2022 into five rounds of data as follows:

Round	# of rows	Description
1	16	First 1/3 of Group Stage
2	32	First 2/3 of Group Stage
3	48	All of Group Stage
4	56	Group Stage and Quarterfinals
5	60	Group Stage, Quarterfinals, and Semifinals

Fitting on Round 1 data

```
set.seed(4747)

# recipe
euro22_recipe <-
  recipe(outcome ~ ., data = r1)

# model
euro22_rf <- rand_forest(mtry = tune(), trees = tune()) %>%
  set_engine('ranger', importance = 'permutation') %>%
  set_mode('classification')

# workflow
euro22_rf_wflow <- workflow() %>%
  add_model(euro22_rf) %>%
  add_recipe(euro22_recipe)

# CV
set.seed(47)
euro22_vfold <- vfold_cv(r1, v=4)

# param
rf_grid <- grid_regular(mtry(range = c(1,6)),
                         trees(range = c(1, 401)),
                         levels = c(10, 9))
```

```
# tuning
euro22_rf_tuned <- euro22_rf_wflow %>%
  tune_grid(resamples = euro22_vfold,
            grid = rf_grid)

# plot
euro22_rf_tuned %>%
  autoplot()
```



```
euro22_rf_best <- finalize_model(
  euro22_rf,
  select_best(euro22_rf_tuned, 'accuracy')
)
euro22_rf_best
```

```
## Random Forest Model Specification (classification)
##
## Main Arguments:
##   mtry = 4
##   trees = 1
##
## Engine-Specific Arguments:
##   importance = permutation
##
## Computational engine: ranger
```

The best model on Round 1 at this seed has parameters `mtry=4` and `trees=1`.

```r
set.seed(4747)
r1_rf_final <-
  workflow() %>%
  add_model(euro22_rf_best) %>%
  add_recipe(euro22_recipe) %>%
  fit(data = r1)

# predict on own data
r1_rf_final %>%
  predict(new_data = r1) %>%
  cbind(r1) %>%
  summarize(accuracy = mean(.pred_class == outcome))
```

```
##   accuracy
## 1    0.625
```

The training accuracy is 0.625.

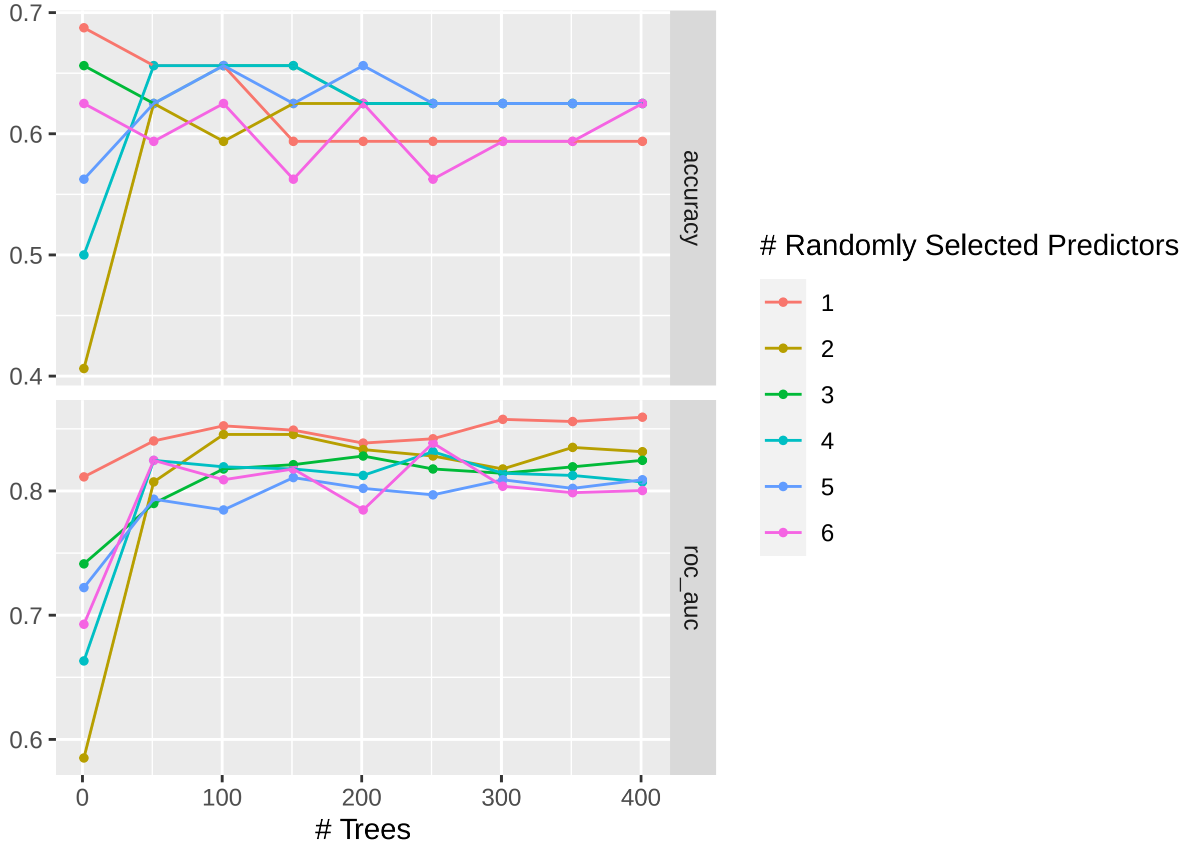
To predict on new data: plug in the dataset of features from the new data into NEWDATA

```r
r1_rf_final %>%
  predict(new_data = NEWDATA) %>%
  cbind(NEWDATA) %>%
  summarize(accuracy = mean(.pred_class == outcome))
```

Fitting on Round 2 data

```r
set.seed(4747)

# recipe
euro22_recipe <-
  recipe(outcome ~ ., data = r2)

# model
euro22_rf <- rand_forest(mtry = tune(), trees = tune()) %>%
  set_engine('ranger', importance = 'permutation') %>%
  set_mode('classification')

# workflow
euro22_rf_wflow <- workflow() %>%
  add_model(euro22_rf) %>%
  add_recipe(euro22_recipe)

# CV
set.seed(47)
euro22_vfold <- vfold_cv(r2, v=4)

# param
rf_grid <- grid_regular(mtry(range = c(1,6)),
                         trees(range = c(1, 401)),
                         levels = c(10, 9))
```

```
# tuning
euro22_rf_tuned <- euro22_rf_wflow %>%
  tune_grid(resamples = euro22_vfold,
            grid = rf_grid)

# plot
euro22_rf_tuned %>%
  autoplot()
```



```
euro22_rf_best <- finalize_model(
  euro22_rf,
  select_best(euro22_rf_tuned, 'accuracy')
)
euro22_rf_best
```

```
## Random Forest Model Specification (classification)
##
## Main Arguments:
##   mtry = 1
##   trees = 1
##
## Engine-Specific Arguments:
##   importance = permutation
##
## Computational engine: ranger
```

The best model on Round 2 at this seed has parameters `mtry=1` and `trees=1`.

```
set.seed(4747)
r2_rf_final <-
  workflow() %>%
  add_model(euro22_rf_best) %>%
  add_recipe(euro22_recipe) %>%
  fit(data = r2)

# predict on own data
r2_rf_final %>%
  predict(new_data = r2) %>%
  cbind(r2) %>%
  summarize(accuracy = mean(.pred_class == outcome))
```

```
##   accuracy
## 1     0.75
```

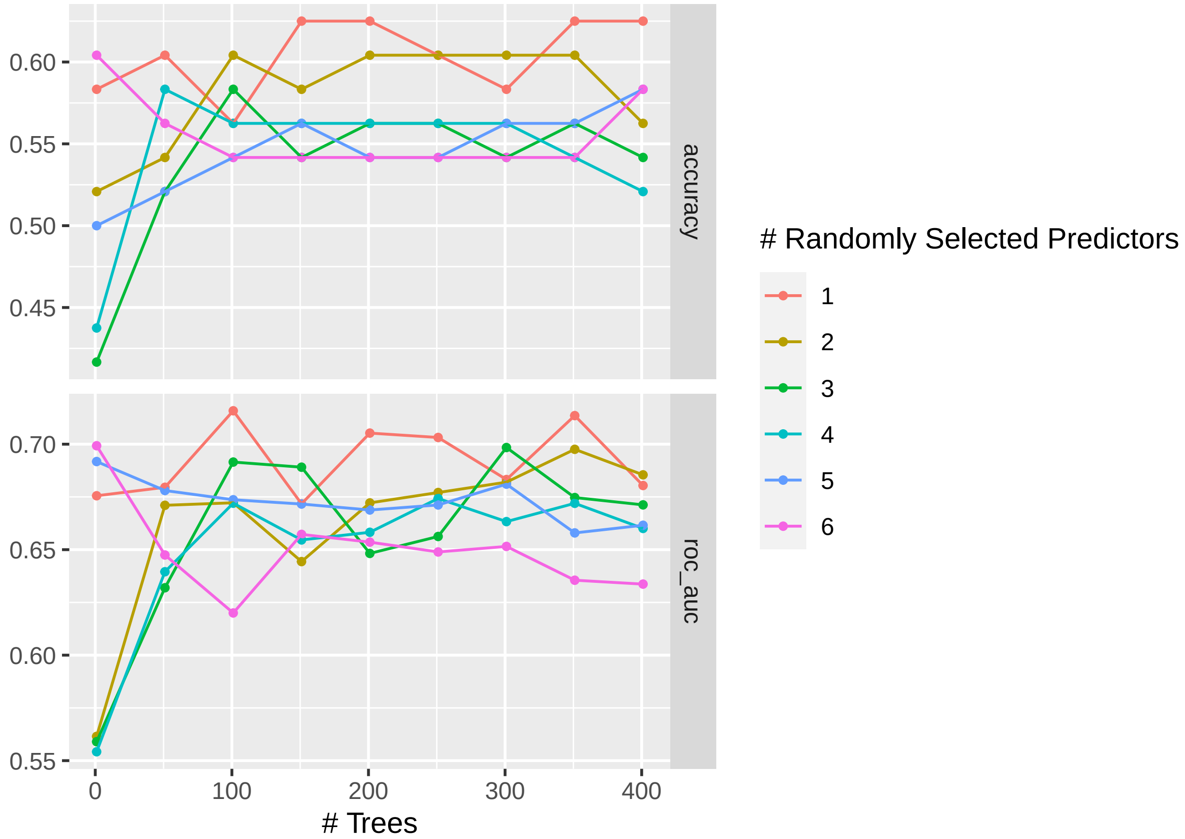
The training accuracy is 0.75.

To predict on new data: plug in the dataset of features from the new data into NEWDATA

```
r2_rf_final %>%
  predict(new_data = NEWDATA) %>%
  cbind(NEWDATA) %>%
  summarize(accuracy = mean(.pred_class == outcome))
```

Fitting on Round 3 data

```
set.seed(4747)

# recipe
euro22_recipe <-
  recipe(outcome ~ ., data = r3)

# model
euro22_rf <- rand_forest(mtry = tune(), trees = tune()) %>%
  set_engine('ranger', importance = 'permutation') %>%
  set_mode('classification')

# workflow
euro22_rf_wflow <- workflow() %>%
  add_model(euro22_rf) %>%
  add_recipe(euro22_recipe)

# CV
set.seed(47)
euro22_vfold <- vfold_cv(r3, v=4)

# param
rf_grid <- grid_regular(mtry(range = c(1,6)),
                         trees(range = c(1, 401)),
                          levels = c(10, 9))

# tuning
euro22_rf_tuned <- euro22_rf_wflow %>%
  tune_grid(resamples = euro22_vfold,
           grid = rf_grid)

# plot
euro22_rf_tuned %>%
  autoplot()
```



```
euro22_rf_best <- finalize_model(
  euro22_rf,
  select_best(euro22_rf_tuned, 'accuracy')
)
euro22_rf_best
```

```
## Random Forest Model Specification (classification)
##
## Main Arguments:
##   mtry = 1
##   trees = 151
##
## Engine-Specific Arguments:
##   importance = permutation
##
```

```
## Computational engine: ranger
```

The best model on Round 3 at this seed has parameters `mtry=1` and `trees=151`.

```
set.seed(4747)
r3_rf_final <-
  workflow() %>%
  add_model(euro22_rf_best) %>%
  add_recipe(euro22_recipe) %>%
  fit(data = r3)

# predict on own data
r3_rf_final %>%
  predict(new_data = r3) %>%
  cbind(r3) %>%
  summarize(accuracy = mean(.pred_class == outcome))
```

```
##   accuracy
## 1    0.875
```

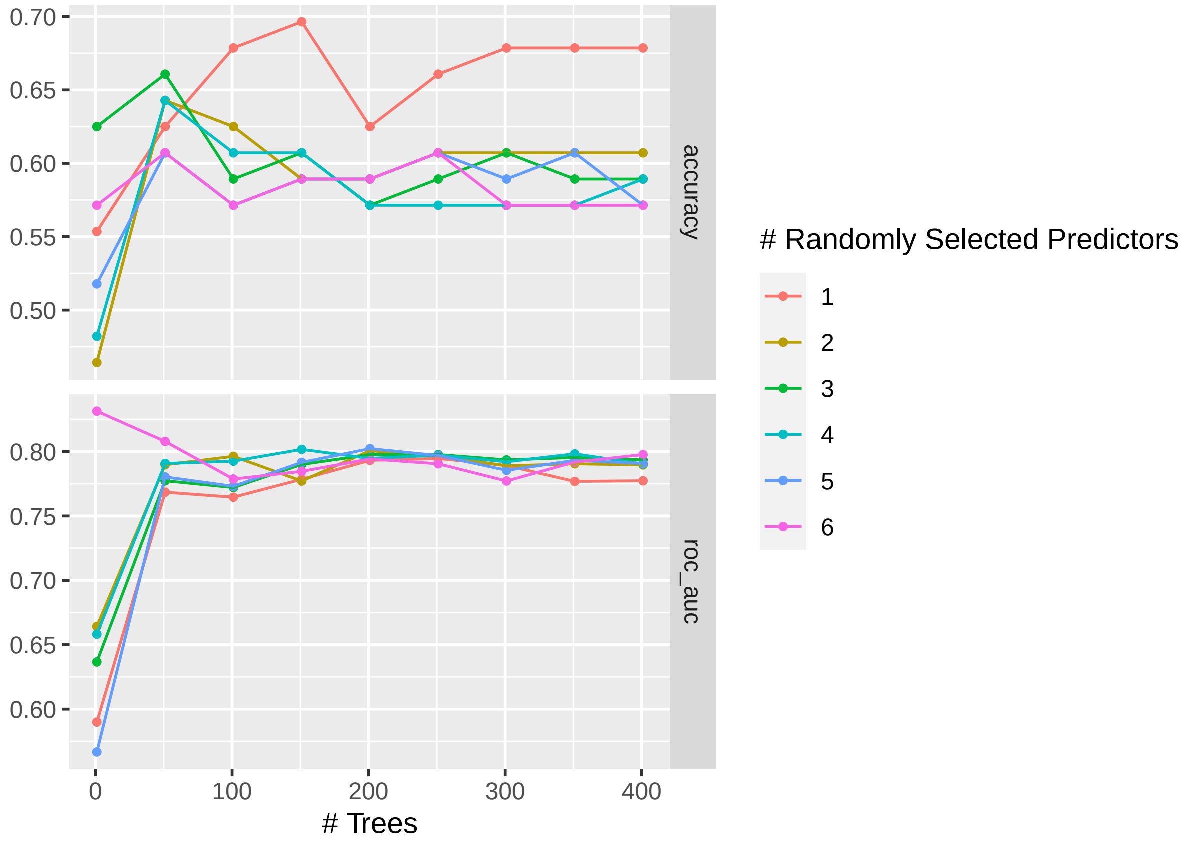
The training accuracy is 0.875.

To predict on new data: plug in the dataset of features from the new data into NEWDATA

```
r3_rf_final %>%
  predict(new_data = NEWDATA) %>%
  cbind(NEWDATA) %>%
  summarize(accuracy = mean(.pred_class == outcome))
```

Fitting on Round 4 data

```
set.seed(4747)

# recipe
euro22_recipe <-
  recipe(outcome ~ ., data = r4)

# model
euro22_rf <- rand_forest(mtry = tune(), trees = tune()) %>%
  set_engine('ranger', importance = 'permutation') %>%
  set_mode('classification')

# workflow
euro22_rf_wflow <- workflow() %>%
  add_model(euro22_rf) %>%
  add_recipe(euro22_recipe)

# CV
set.seed(47)
euro22_vfold <- vfold_cv(r4, v=4)

# param
rf_grid <- grid_regular(mtry(range = c(1,6)),
                          trees(range = c(1, 401)),
                          levels = c(10, 9))

# tuning
euro22_rf_tuned <- euro22_rf_wflow %>%
  tune_grid(resamples = euro22_vfold,
            grid = rf_grid)

# plot
euro22_rf_tuned %>%
  autoplot()
```



```
euro22_rf_best <- finalize_model(
  euro22_rf,
  select_best(euro22_rf_tuned, 'accuracy')
)
euro22_rf_best
```

```
## Random Forest Model Specification (classification)
##
## Main Arguments:
##   mtry = 1
##   trees = 151
##
## Engine-Specific Arguments:
##   importance = permutation
```

```
##
## Computational engine: ranger
```

The best model on Round 4 at this seed has parameters `mtry=1` and `trees=151`.

```
set.seed(4747)
r4_rf_final <-
  workflow() %>%
  add_model(euro22_rf_best) %>%
  add_recipe(euro22_recipe) %>%
  fit(data = r4)

# predict on own data
r4_rf_final %>%
  predict(new_data = r4) %>%
  cbind(r4) %>%
  summarize(accuracy = mean(.pred_class == outcome))
```

```
##   accuracy
## 1    0.875
```

The training accuracy is 0.875.

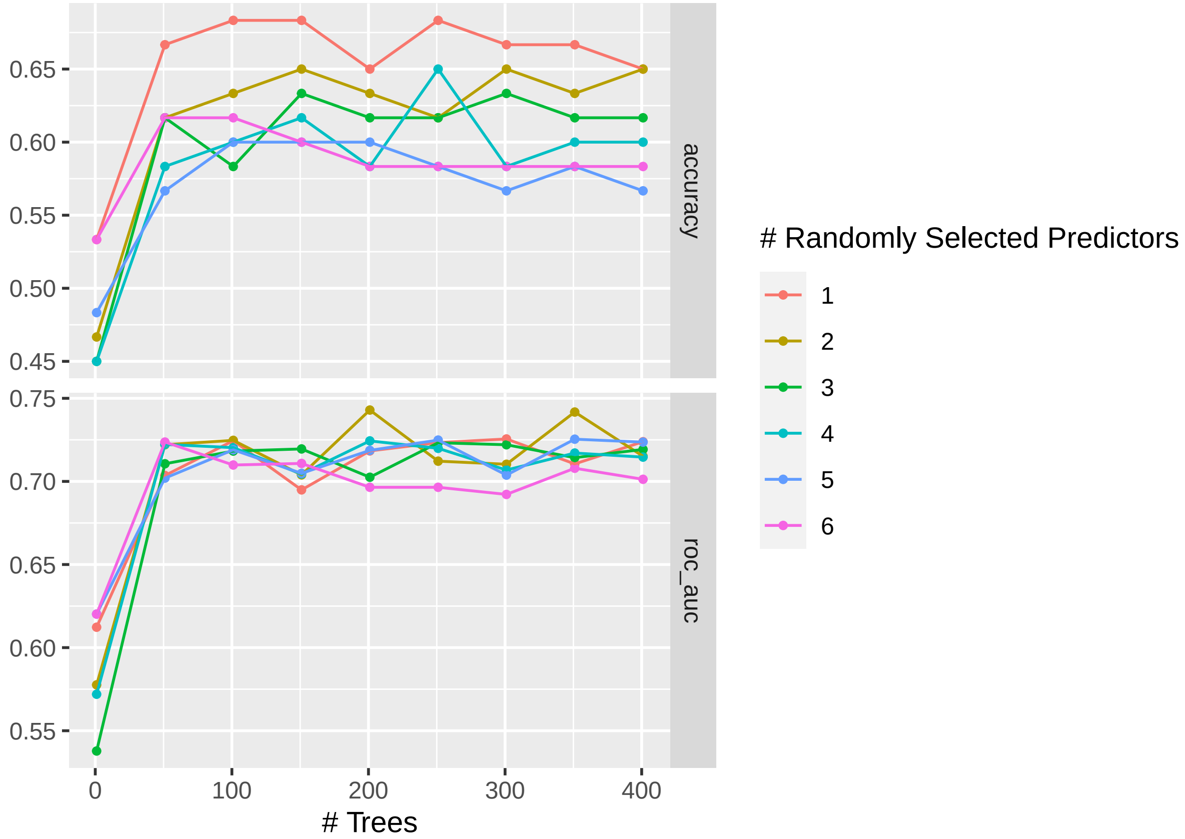
To predict on new data: plug in the dataset of features from the new data into NEWDATA

```
r4_rf_final %>%
  predict(new_data = NEWDATA) %>%
  cbind(NEWDATA) %>%
  summarize(accuracy = mean(.pred_class == outcome))
```

Fitting on Round 5 data

```
set.seed(4747)

# recipe
euro22_recipe <-
  recipe(outcome ~ ., data = r5)

# model
euro22_rf <- rand_forest(mtry = tune(), trees = tune()) %>%
  set_engine('ranger', importance = 'permutation') %>%
  set_mode('classification')

# workflow
euro22_rf_wflow <- workflow() %>%
  add_model(euro22_rf) %>%
  add_recipe(euro22_recipe)

# CV
set.seed(47)
euro22_vfold <- vfold_cv(r5, v=4)

# param
```

```
rf_grid <- grid_regular(mtry(range = c(1,6)),
                        trees(range = c(1, 401)),
                        levels = c(10, 9))

# tuning
euro22_rf_tuned <- euro22_rf_wflow %>%
  tune_grid(resamples = euro22_vfold,
            grid = rf_grid)

# plot
euro22_rf_tuned %>%
  autoplot()
```



```
euro22_rf_best <- finalize_model(
  euro22_rf,
  select_best(euro22_rf_tuned, 'accuracy')
)
euro22_rf_best
```

```
## Random Forest Model Specification (classification)
##
## Main Arguments:
##   mtry = 1
##   trees = 101
##
## Engine-Specific Arguments:
```

```
##   importance = permutation
##
## Computational engine: ranger
```

The best model on Round 5 at this seed has parameters `mtry=1` and `trees=101`.

```
set.seed(4747)
r5_rf_final <-
  workflow() %>%
  add_model(euro22_rf_best) %>%
  add_recipe(euro22_recipe) %>%
  fit(data = r5)

# predict on own data
r5_rf_final %>%
  predict(new_data = r5) %>%
  cbind(r5) %>%
  summarize(accuracy = mean(.pred_class == outcome))
```

```
##   accuracy
## 1  0.86667
```

The training accuracy is 0.867.

To predict on new data: plug in the dataset of features from the new data into NEWDATA

```
r5_rf_final %>%
  predict(new_data = NEWDATA) %>%
  cbind(NEWDATA) %>%
  summarize(accuracy = mean(.pred_class == outcome))
```